Multimedia

multiple — agents to communicate something

∴ Multimedia = usage of multiple agents to communicate information.

these media/agents could be of the form of text, audio, video, animation, etc.

Media can be categorized based on few criteria

→ perception media → how we percieve ~~inf~~ information

→ visual media (text, images, etc)

→ auditory media (music, sounds, etc)

→ Representation media → how we store the information in a computer?

ex → text in ascii format

→ audio files in pcm

→ images in .png, .jpeg, .jpg etc

→ videos in .mp4, .mpeg, etc

→ presentation media → how to present/display/show/express the media from the computer or to the computer.

→ physical means used by computer or systems to reproduce info to humans.

eg :- O/P paper, monitors, loud speakers, etc.

I/P → mouse, cameras, microphone.

Storage media :- where the info is stored.

→ Reffers to various physical means of storing the data.

Eg:- CD Rom, dvd, usb, etc.

Transmission media :- How to transmit the data

→ Reffers to the physical means of transferring the data from one place to another (or) one system to another

eg:- optical fiber, coaxial cable, twisted pair, etc

Categories of Multimedia

→ linear and non-linear media :-

- linear content progresses without any navigation (eg cinema)
- non-linear content progress with navigation based on user interactivity to control. (eg computer game, website, etc)

→ interactive & non interactive multimedia

→ realtime & recorded multimedia

Elements of multimedia

→ Text
→ graphics
→ Audio
→ Video

graphics → Bitmaps → Real images captured from devices such as cameras, etc, Not editable, large size

graphics → Vectorgraphics
→ drawn on computer
→ editable
→ small size

$\frac{512\times512\times1}{8}$

1M = 1000KB ×8 bits

$\boxed{1MB = 1024\ KB}$

1 Bit per pixel

1 1

n 512×512

512×512 pixels

$\frac{512\times512}{8}$ → bit n

8 bit → 1 by

$x = \frac{512\times512\times8}{8}$

$512\times\cancel{512}\times2^9$

2^9 MB

$\frac{512\times512}{1024\times1024}$ $\frac{1}{4}$

~~512~~

Images

→ Represented as a (Bit Map)
↓
a grid of pixels organized as a 2D array.
width height

→ each pixel has a bit depth
↓
represents the number of bits assigned to each pixel

→ if bit depth = 1 → only 1 pixel is required to represent the pixel.

bit depth = 8 bits (greyscale image)

" " = 24 bits (rgb image)

Audio

→ characterized by a sampling rate in hertz, which gives no. of samples per second.

→ a sample is an individual unit of audio information.

→ sample size lies from 8 bits → 16 bits.

→ CD Quality audio requires 16 bit sampling at 44.1 KHz

→ 16 bits are required for a sample and 44.1×10^3 samples/sec (or) 44.1×10^3 samples per second and each sample is 16 bit long.

Video

→ It's represented as a ~~series~~ sequences of images.

→ properties :- width, height, pixel depth

→ parameter :: frames per second (is the frequency at which the consecutive images (frames) are captured or displayed

Multimedia Systems

-> A multimedia system is a system which is capable of processing/ storing/ generation/ manipulation and rendition multimedia data and application

characteristics of M systems

→ computer controlled

→ integrated → multiple media types

→ represented digitally

→ interactiveness

Challenges in MS

→ Distributed networks → transfer of multimedia from one system to another, transferring large files of multimedia is not feasible with network bottlenecks/overheads.

→ temporal relationship b/w data

→ render different types of data from different media at the same time - continuously.

→ sequencing within the media :- playing frames in correct order (intra media scheduling)

→ synchronization :- inter-media scheduling, synchronization b/w two different types of media.

Desirable features of MS

(i) High processing power → to deal with large data and process it.

(ii) Capable file system → to store data and retrieve in real time

(iii) Special hardware/software → eg. RAID technology

(iv) Data representation :- file ~~support~~ type formats that support multimedia should be easy to handle different processes like compression/decompression in real time

(v) Efficient I/O :- faster I/P and O/P to allow real time recording as well as playback of data
eg:- Direct to disk recording systems.

(vi) Special OS :- It is required inorder to ~~the~~ cater the fast I/O, processing of multimedia in real time.

(vii) Network support.

Component of MS

1) content production :- (i) capturing data from different ~~media~~ instruments in the digital format, these include camera, microphone, etc.

(ii) Once media elements are in their digital representations, they are further combined to create a Coherent, interactive media using S/w applications and H/w elements

(iii) the content created is then stored into the disk.

2) compression and Storage (i) deals with the compression of multimedia content which reduces the size of the media file

(ii) encode it and store

3) Distribution :- (i) to distribute the media accross the network

Text

→ A glyph is a graphical representation of a character's shape where a character may be represented as many glyphs.

→ font size is measured in points, and one point is approx 1/72 of an inch. 0.0138

Digital Representation of text :-

(i) ASCII

(ii) Unicode

ASCII

→ normal Ascii — 7 bits

0 →

$\therefore 2^{7} - 1 = 127$

$\therefore 0 - 127 = 128$ possible combinitons

→ extended Ascii — 8 bits

$\therefore 2^{8} - 1 = 0 - 255$

Unicode (unique, Universal and Uniform character encoding)

→ ASCII is a subset of unicode.

A → 65 → 1000001

UTF-8 → uses 1 byte for first 128 Characters and go upto 4 bytes for other characters

UTF-16 → uses 16 bits to represent each character, hence capable of representing 2^{16} characters

UTF-32 → uses 32 bits to represent each characters, hence capable of representing 2^{32} characters.

Problems with UTF-16 and UTF-32

→ waste of lot of space

for example UTF_{32} representation of A is

$$\underbrace{00\ldots}_{\text{these many waste spaces}}01000001$$

these many waste spaces, similar things follow for UTF-16

→ Many older computers interpret 8 zeros consecutively as a null which is end of the string of characters. hence they will not represent 'A' if encoded in UTF-16 or UTF-32.

UTF-8 encoding

→ upto 128th character, the regular ASCII value is use so for example A is 1000001

→ for after 128th character, UTF-8 seperates the code into two bytes

$[110\underbrace{xxxxx}_{\text{to be filled}}][10\underbrace{xxxxxx}_{\text{to be filled}}]$

no. of bits to be filled = 11
∴ It can encode upto 2^{11}
= 2048

Ex:-
325 $\xrightarrow{\text{binary}}$ 000 | 01 0 0 0 1 0 1

2	325	
2	162	1
2	81	0
2	40	1
2	20	0
2	10	0
2	5	0
2	2	1
2	1	0
		1

↓ UTF-8 encoding

[11000010] [10000101]

→ after 2048th number
- one more 1 is added to the 3rd byte

[1110 xxxx] [10xxxxxx] [10xxxxxx]

no. of bits to be filled = 16 ∴ It can encode upto 2^{16} numbers

→ after 2^{16}th numbers
- one more 1 is added to the 4th byte

[11110 xxx] [10 xxxxxx] [10xxxxxx] [10xxxxxx]

no. of bits to be filled = 21

∴ it can hold upto 2^{21} numbers

Text file formats

-> •txt -> no styles attached

-> simple text

Advantages :- (i) it's compatible with all machines

(i.e) the end result will be same.

(ii) It's fast and flexible

(iii) file size is less as compared to •rtf

-> •rtf -> styles attached

-> created by Microsoft

Advantages :- (i) ~~reado~~ Readability

(ii) 90% of text editors support it

(iii) styled, formatted text is shown to user

disadvantages:- (i) file size is large

(ii) unsupported editor displays gibberish text

Digital Image Representation

→ The most common form of representing natural images and other forms of graphics is a bit map
↓ refers to
how a given pattern of bits maps to a specific color.

→ info is stored in the form of grid of pixels, which has fixed width and height, and can store various ranges of colours based on image type

Image Data types

1 Bit Images

→ consists of ~~one and~~ 0 and 1 bits only
→ ~~mono~~ called sometimes 1 bit monochrome image.
→ reffered to as a binary ~~img~~ image.
→ A 640 × 480 pixels monochrome image requires = 640×480/8 = 38.4 KB

8 Bit Gray-level images

→ each pixel has a gray value b/w $0 - (2^8 - 1)$ (i.e) 0 – 255.
→ Each pixel is represented by a single byte
→ Entire image is a 2-D ~~image~~ array of pixels
such an array is a bitmap

→ a 640 x 480 grayscale image requires
640 x 480 = 307,200 KB.

24 Bit Color Images (RGB image)

→ each pixel is represented by three bytes.

~~each pixel is~~

→ each pixel's value lies in the range 0-255, this format supports 256x256x256 colours

→ RGB images is stored as ~~an~~ 32-bit images, the extra bit represents the transparency flag.

Resolution

→ measure of how finely a device displays graphics with pixels.

→ dpi (dots per inch) → number of dots within one inch of an image printed by a printer.

→ ppi (pixels per inch) → number of pixels contained within one inch of an image displayed.

8 bit colour Images

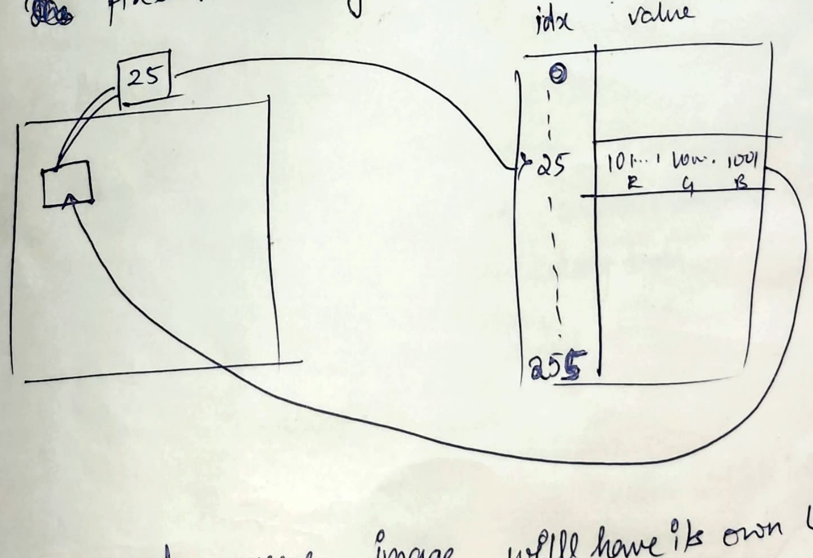
→ If space is a concern we can use this type of images rather than the 24 bit images.

→ many systems can ~~store~~ utilize the colour stored with only 8 bits (256 colors)

→ It uses a concept called look up table to store colour information.

→ The images doesn't store the colour, but instead stores a value which is an index into the look up table which ~~map the~~ is mapped with a 3-byte values that specify the colour for a pixel.

→ ~~the~~ pixel value only contains an index



→ each and every image will have its own LUT and ~~while transmission the LUT~~ the LUT is stored in the image file header

Image file formats

8-bit GIF (graphics interchange format)

→ using this we can represent 8 bit colour images.

→ limited to 8 bit (256 colours) images only.

→ It supports interlacing - successive display of pixels in a widley spaced rows by a 4-pass process

→ whenever we open an image it doesn't show up completely, it comes in parts/lines.

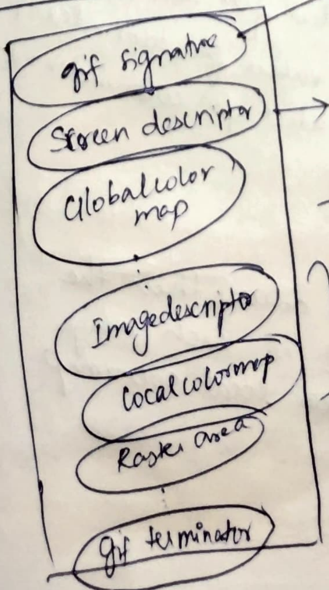
∴ here in the 1st pass every 8th line is shown
2nd pass every 4th line is shown, 3rd pass - 2nd line,
4th pass all other lines.

GIF

Gif 87a $\xrightarrow{\text{extended}}$ GIF 89a

→ supports animation via a graphics control block
→ supports control over delay time
→ supports transparency index

Gif file format



→ 6 bytes header which indicates that it's a gif file

→ 7 bytes set of flags

→ since Gif is a collection of images each image will have its own Image descriptor, local colormap, look up table raster area,

Screen descriptor

→ It consists of a set of attributes which is common to all the images in the file

Bits 7 6 5 4 3 2 1 0		Byte	
Screen width		1	width in LSB first
		2	
Screen height		3	height in LSB first
		4	
m | cr | 0 | pixel		5	→ no of bits per pixel in image. (bit depth)
Background		6	→ gives the index byte for the background colour.
0 0 0 0 0 0 0 0		7	for future extentions

m = 0 represents the global colour map is not present
m = 1 " " " " " " present

-

GIF color map

Bits 7 6 5 4 3 2 1 0	Byte	
Red	1	Red value for color index = 0
Green	2	Green value for color index = 0
Blue	3	Blue value for color index = 0
R	4	Red value for color index = 1
G	5	green value for color index = 1
B	⋮	Blue value for color index = 1
	256	

→ if all the images follow the same colours then the Global color map is used. But if each image has its own colours then the local colourmap is used.

GIF Image descriptor

→ It's a 10 bytes attribute

Byte	7	6	5	4	3	2	1	0	
1	0	0	1	0	1	1	0	0	→ Image seperator character indicates at which places we have images
2									
3	Image left								→ starting point of the image from the left
4									
5	image top								→ from the top of the screen where does the image start
6	Image width								→ width of image in pixels
7									
8	~~Image height~~								→ ~~width of~~ height of Image in pixel
9									
10	m	i	0	0	0	pixel			

m=0 use global color map, ignore pixel

m=1 use local color map, use pixel

i=0 Image is formatted in sequential order

i=1 Image is " " interlaced order

pixel bits per pixel.

Raster Area

→ stores the actual image information.

→ Stores the pixel color index values that make up the image in a sequential manner.

→ stored from left to right in each row and top to bottom.

→ if i=1, then image display follows a 4 pass process.

GIF terminator

→ shows that end of the GIF file by a character 0X3B or ;.

JPEG

→ current standard for image compression

→ It takes the limitation of human vision system as an advantage to acheive high rates of compression.

→ as human eye can't see every pixel detail,

→ if many changes occur within a few pixels, we refer to that segment as having high spatial frequency.

→ since we are throwing away some information, hence its a lossy compression.

→ It allows the user to set a desired level of quality compression ratio. (input divided by output)

PNG (Portable network graphics)

→ superseed GIF.

→ supports 16 bits per pixel in each color channel

∴ 48 bit color – a large increase.

→ gamma correction for correct display of colour and alpha channel for transparency.

→ It progressively displays pixels in a 2D fashion by showing a few pixels at a time over ~~seven~~ 7 passes through each 8×8 block.

→ It supports both lossy & lossless compression

→ performance is better than JPEG and 8 bit images

TIFF (Tagged image file format)

→ we can have additional information in the form of tags.

→ format specifier tag tells us what type of compression is used in storage of the image

→ It can store different type of images, 1-bit, grascale, 8 bit color, 24 bit RGB etc

→ TIFF is a lossless but a JPEG tag allows it to opt the lossy compression

EXIF (Exchange image file)

→ image format for digital cameras.

→ many more tags than TIFF, which can store info like picture taking condition, exposure, etc can be used by colour-correction techniques to improve the picture quality

→ It is encorporated into the JPEG software in most digital cameras.

PS

→ It's an important language for typesetting and many high end printers have a postscript interpreter built into them

→ Vector based language, rather than pixel based language

→ Can be used in other ~~file formats~~ documents via
- EPS

→ Postscript files are stored as ASCII

→ No compression in PS

PDF

→ uses LZW file compression

Color Model

→ It's a orderly system for creating a whole range of colours from a small set of primary colours

→ Two models
- → additive colourmode (eg RGB)
- → subtractive color model (eg CMY)

→ Additive colours models use light to display colour while subtractive use printing inks

→ Additive colour modes, eg: computer monitor, etc

→ subtractive are printers, etc.

→ colours perceived in additive models are a result of transmitted light

→ colours perceived in subtractive model are a result of reflected light

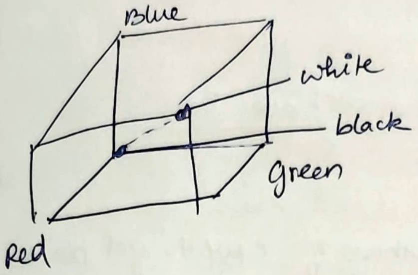
→ RGB model (computer display)

→ CMYK mode (printing)

→ in RGB model we add colours to go from Black to white

→ in CMYK model we go from white to black

RGB model



→ grey shades are represented along the diagonal

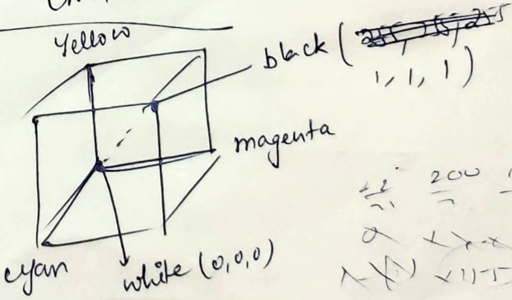
→ each point in the colour cube can be represented as a sum of weighted vector sum of the primary colours R, G, B

$$C(\lambda) = R\mathbf{R} + G\mathbf{G} + B\mathbf{B}$$

where $R, G, B = \{0, \ldots, 1.0)$

CMY and CMYK model

→ CMY stands for cyan magenta yellow



RGB to CMY

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

CMY to RGB

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RGB and CMY are complement of each other

$RGB = (x, y, z)$

$\updownarrow$

$CMY = (255-x,\ 255-y,\ 225-z)$

CMYK → store extra black ink

→ mixing equal proportions of CMY to get black is often not possible as it might endup to a brown ink. So simple approach is to store the black ink itself

→ the process of removing the three colour mix to form black and adding it back as a real black is known as undercolor removal.

$$K = \min\{C, M, Y\}$$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} \Rightarrow \begin{bmatrix} C-K \\ M-K \\ Y-K \end{bmatrix}$$

Video

→ a sequence of discrete images shown in quick succession.

→ each image in the video is called a "frame" which is represented as a matrix of pixels defined by w, h, pixel depth

→ other two properties

(i) frame rate :- The rate at which the images are shown

units :- frames/second.

Eg : 25 frames/second means 25 images/frames are shown in a second.

(ii) scanning format < interlaced, progressing

→ if the frame rate is too slow, the human eye perceives an uneveness of motion called flicker.

→ digital video is 3D as a 2D image is changing over time but an analog video is converted to a 1D signal of scan lines.

Analog video and television

→ It is scanned as a 1D signal in time

→ The 1D signal captures the timevarying image intensity

→ Television broadcasts this scanned analog information from the broadcast station to all the users.

→ The broadcast of the ~~and~~ analog video requires some other processes like → (i) YUV color space conversion

(ii) interlaced scanning.

Conversion of RGB to YUV

→ Video frames / Images are represented using RGB model.

→ The RGB is used by the televisions to display and render the video signal

→ The main problem occurs while transmission of the RGB model, as during transmission there may be some losses and rendering of the signal back to frames loses some information regarding color, etc.

→ hence the researchers came up with YUV format based on the fact the human visual system is tolerant to color losses than the intensity losses.

∴ Y (stores the information about the intensity) → sent over high band width so information will not be lost and we will be able to see what is in the video.
↓
luminance

UV (stores the information about the colours) → sent over low band width
↓
Chrominance or Chroma

Analog Video Scanning

→ Video is scanned as a 1D signal, where each line is interspaced with horizontal and vertical syncs

→ for horizontal sync, a small voltage offset from zero is used to indicate black and another value such as zero to know about start of a new line

→ for vertical sync, it's carried out by cycles in the power outlet. for NTSC it's 60Hz meaning after ever $1/60^{th}$ of the second the electron gun is reset to get a new frame.

interlaced scanning

→ first all the odd lines are scanned and then the even lines.

→ therefore it's ~~[illegible]~~ unacceptable and has occasional flickers ~~[illegible]~~ which is due to the fact that the video is captured at different moments in time

Progressive scanning

→ Video is better captured in this scanning format as it eleminates the occasional flicker

NTSC (National Telivision System Comittee)

→ 4:3 aspect ratio (ratio of pictures width to height)

→ 525 scan lines per frame at 30 fps.

Pixelclock

→ no fixed horizontal resolution, therefore we must decide how many times to sample the signal for the display.

→ mostly used in America and Japan

PAL (~~Progressive~~ Phase Alternating line)

→ TV standard

→ 625 scan lines perframe at 25 fps.

→ 4:3 aspect ratio

→ uses YUV colour model with 5.5 MHz to Y and 1.8 MHz each to U and V

→ to improve picture quality, the chroma signals have alternate signs (+U and -U) in the success scan lines (hence phase alternating line (PAL)

SECAM

→ 625 scanlines for frame at 25 frams/sec

→ 4:3 aspect ratio

→ U → 4.25 MHz

V → 4.41 MHz

→ only one of the U or V signals is sent on each scan line

525	625	625
30	25	25
4:3	4:3	4:3

Types of Video Signals

→ **composite video**.

→ luminance, chrominance, syncronization and blanking pulses all together are transmitted over a single signal.

→ This done to reduce bandwidth and acheive real time transmission (advantage)

disadvantages

→ interference b/w chrominance and luminance will be there and worsen on a weak signal

→ fluctuating colours, false colours and intensity variations are seen.

S video (super video / (Y/C) video)

- each of the luminance and chrominance signals are transmitted seperately to acheive superior quality. (but will take more bandwidth)
- the luminance (Y) carries brightness info and the chrominance (C) carries the color info.
- the U and V are combined into C and transferred as a single one, at display time again the C is sperated into U and V
- seperating Y and C reduces the problems caused by interferance.

Component video

- keeps Y, U, V as seperate components and transmit them in three signals
- bandwidth required is high
- visual quality is high.
- seperating them into three different signals prevents the interference.

Digital Video

- Advantages to digital representation of a video

→ it becomes easier to store them in harddrives

→ once stored we can perform edits on it using different multimedia applications.

→ non linear video editing is simple

→ repeated recording and editing doesn't degrade the image quality.

→ ease of encryption and better tollerance to channel noise

YUV subsampling schemes

→ In analog video, subsampling is achieved by given more than half bandwidth to Y and other bandwidth to UV

→ In digital video, subsampling can be done by reducing the number of bits used for color channels on an average.

4:4:4 ⇒ all the Y, U, V are given equal chance

→ 24 bits per pixel

4:2:2 ⇒ chrominance information stored alternatively

→ $\frac{24+8}{2} = 16$ bits per pixel

4:1:1 → chrominance information stored for every fourth pixel in a row

→ $\frac{24+8+8+8}{4} = 12$ bits per pixel.

$4:2:0 \Rightarrow \frac{8\times 4 + 16}{4}$ ~~$\times 8 \times 8$~~ $= 12$ bits/pixel

why dowe need subsampling ?

-) inorder to reduce the data size by selecting a subset of the original data.

Digital video standards

· CCIR and ITU-R → international

↓

consultative commitee for international Radio

→ NTSC has 525 scan lines, each having 858 pixels

NTSC uses 4:2:2 subsampling

∴ the average bits per pixel = 16 bits/pixels

hence the data rate = $\underbrace{\underbrace{\underbrace{525 \times 858}_{\text{no. of pixels in oneframe}} \times 16}_{\text{no. of bits in one frame}} \times 30}_{\text{no. of bits in 30 frames as NTSC transfers data at 30Hz/fps}}$

= 216 Mbps
↓
Megabitspersecond

$= \frac{216}{8}$ MBs
megabytes persecond.

CIF (common interchange format)

→ width and height divisible by 8 which is a requirement for digital encoding algorithms.

→ It's a compromise b/w NTSC and PAL as it adopts the frame rate of NTSC and half the number of active lines in PAL.

HD-TV

- The width is increased so more pixels per frame.
- aspect ratio is 16:9
- increased resolution provides a clearer and detailed picture. In addition to this progressive scan and higher frame rates result in a picture with less flicker and better rendering of fast motion.
- while transferring the video as the resolution is increased the bandwidth requires is also increased. for this reason MPEG2 video compression is used for HD-TV
- But still its requires huge bandwidth in order to transmit numerous channels.
- 60fps

UHD-TV

- resolution is even more high than HDTV
- Supports 4K : 2160p (3840 pixels per scanline and 2160 scan lines)

  8K : 4320p (7680 pixels per scanline and 4320 scanlines)

- 16:9, bit depth = 12
- subsampling can be either 4:2:2 or 4:2:0
- 120 fps

Digital display Interfaces

→ VGA (analog video interface)
→ DVI
→ HDMI
→ Display port

digital display interfaces

DVI

→ used to connect computer's video card to the monitor
→ supports both analog and digital
→ It makes backward compaitable with VGA
→ allows a maximum 16:9 screen resolution of 1920X1080 at 60Hz

HDMI (High Definition Multimedia interface)

→ not compatible with analog signals
→ supports RGB as wells a YUV
→ digital audio + digital video (supports)
→ HDMI 2.0 supports 4K resolution at 60fps

Display port

→ It will send information in the form of packets like internet

→ micro packets which can embed ~~time~~ the clock signal

→ It can transmit both audio or video or both simultaneously

→ It has slightly more bandwidth.

→ It's royality free, while HDMI charges an annual fee to the manufacturers.

$d = f b / Z$
disp
dept
distance b/w the two cam
focal lens

3D Video and TV

3D movie using coloured glasses

A

→ glasses tinted with complimentary colours, usually red on the left and cyan on right. This technique is called Anaglyph 3D.

→ left image is filtered to remove Blue and green and the right image is filtered to remove red.

→ when viewed using these glasses, each of the two images reach the eye it's intended for, revealing an integrated stereoscopic image.

→ then the visual cortex of the brain fuses this into the perception of a 3D scene.

→ Even though the Anaglyph 3D movies are easy to produce due to the colour filtering, the colour quality is not good.

3D movies using circularly polarized glasses

→ RealD cinema system

→ movie goers are required to wear the polarized glasses

→ The left and the right pictures are polarized in different directions and then they are projected and superimposed on a screen

→ a user wearing the glasses are polarized accordingly and one of the two pictures is passed through and others are blocked.

3D TV with shutter glasses

→ It's used in TV's for home entertainment.

→ the liquid crystal layer on the glasses become opaque when some voltage is applied

→ Then the glasses are synchronoized with the TV, which alternatively shows different images in a time sequential manner

for eg (120 Hz for left and 120 Hz righ)

$\frac{8 \times 4 + 16}{4}$

$\frac{16 + 4 \times 8}{4} = 12$

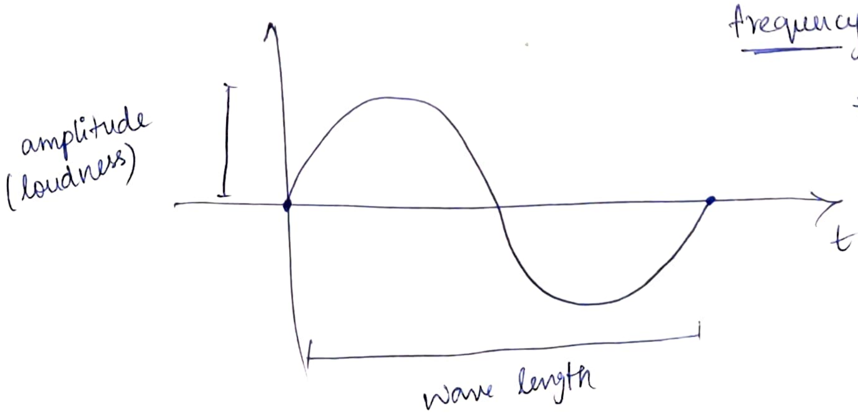
Audio

-> It's crucial for multimedia presentations
-> It's the simplest type of multimedia
-> some differences b/w audio vs video, dropping some video frames is ok but if you drop some audio info, all the sense will be lost.

Sound

-> It's a wave phenomenon in which molecules of air is being compressed and expanded under the action of some physical device
-> It's a pressure wave
-> It's a continuous signal.
-> we need to convert audio info to digitized form
-> even though pressure waves are longitudinal, they still have ordinary wave properties and behaviours such as reflection, refraction, etc.
-> a signal can be decomposed into a sum of sinusoids

Characteristics of Sound



frequency -> is the rate of vibration (referred to as pitch)
-> cycles/second
or
Hz

Digitization

→ sound is a 1D signal

→ to fully digitize ~~the~~ a sound wave we have to sample it both in time and amplitude domain.

→ Sampling → measuring the quantity we are interested in, at evenly spaced intervals

→ The rate at which sampling is done is called sampling rate. (no. of samples collected per second)

→ for audio typical sampling rates are from 8KHz to 48KHz

→ the human can hear from about 20 Hz to 20KHz

Nyquist Sampling rate

→ to preserve the full information in the signal its necessary to sample at least twice the max frequency of the signal, that is known as the Nyquist ratio

→ If we sample the signal at a lower frequency than nyquist rate, when the signal is converted back to a continuous time signal, it will exhibit aliasing

↓

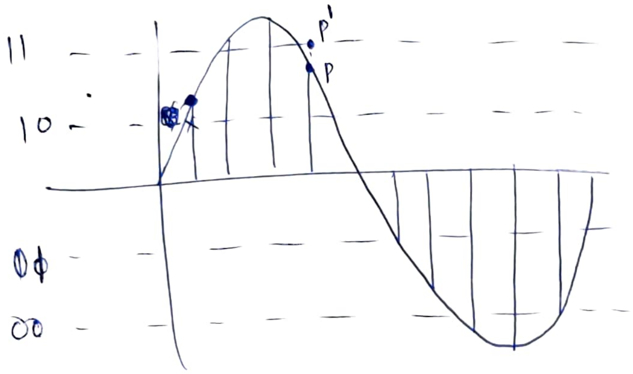
presence of unwanted components in the reconstructed signal.

Quantization

→ It's sampling the analog signal in the amplitude dimension.

→ in case of Quantisation we can do nonuniform sampling

2 bit Quantization



Quantization error $= p' - p$

→ so if we increase the level of quantization then we can decrease the Quantization error

→ In soundwaves, the uniform Quantization rates are 8 bits and 16 bits

2^8 levels $\quad 2^{16}$ levels

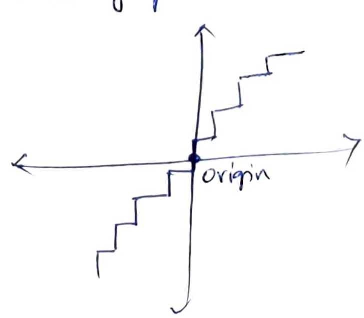
→ Quantization error will always lie in the range 0 to half of the discretization step.

(or)

the error can have a max error of one half of the discretization step.

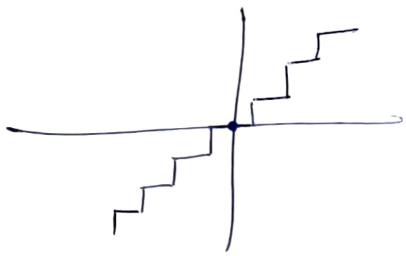
Uniform Quantization

• → Mid Rise :- origin lies in the middle of the rising part of the stair case like graph



~~Mid-Thread~~

Mid-tread :- origin lies in the middle of a tread of the staircase like graph



Qunatization error :- difference in the actual value and the nearest quantized value.

Non-Uniform Quantisation

→ the stepsize is not constant

→ We have to use this quantization where there is a lot of variation in the amplitude, i.e. amplitude is high. can be expressed as a crest factor

crest factor = peak ~~value~~ value of signal / rms value of signal

→ It can be acheived by Companding

compressing + expanding

compressing → done at sender side

expanding → done at receiver side

μ law and A law companding

μ law $\rightarrow$

$$r = \frac{\text{sign}(s)}{\ln(1+\mu)} \ln\left\{1 + \mu\left[\frac{s}{S_p}\right]\right\} \quad \left|\frac{s}{S_p}\right| \le 1$$

$\mu = 0$ no compression or expansion
$\mu \uparrow \leftrightarrow$ more compression

A law

$$r = \begin{cases} \frac{A}{1-\ln(A)}\left(\frac{x}{x_0}\right) & \left|\frac{s}{S_p}\right| \le \frac{1}{A} \\ \frac{\text{sign}(x)}{1+\ln(A)}\left[1 + \ln A\left|\frac{s}{S_p}\right|\right] & \frac{1}{A} \le \left|\frac{s}{S_p}\right| \le 1 \end{cases}$$

$$\text{where } \text{sign}(s) = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{otherwise} \end{cases}$$

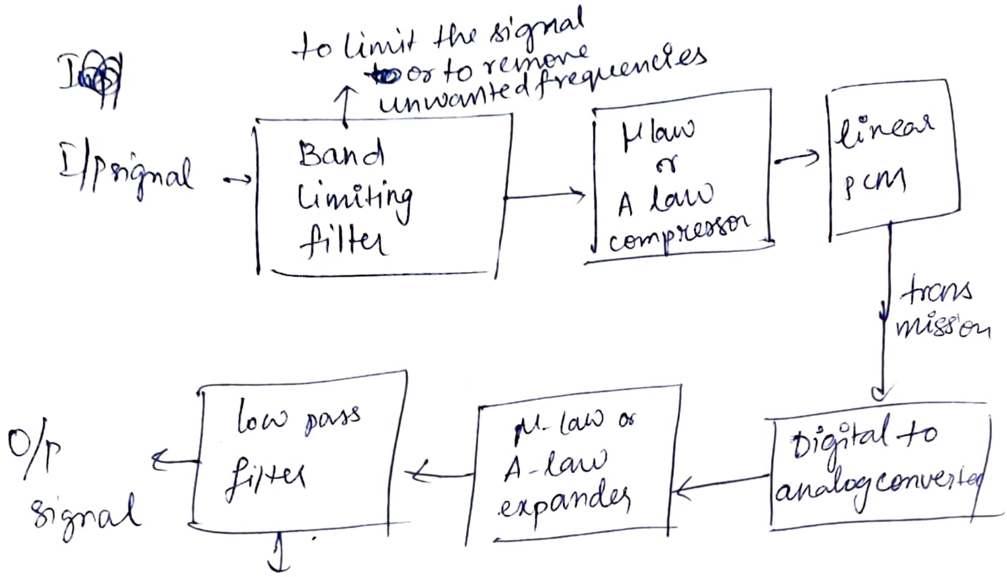
Pulse code Modulation (PCM)

→ is the processing of modifying one or more parameters of a carrier signal in accordance with the instaneous values of the message signal.

→ Basic techniques for creating digital signals from analog ones is sampling and quantization.

→ PCM is a formal term for sampling and quantization.

Basic elements of a PCM



after expanding some frequencies may go out of range, so to remove those we use this filter

Differential pulse code modulation (DPCM)

→ Based on predictive coding

→ Audio is often stored not in simple PCM but in a form that exploits differences.

$$\hat{f}_n = \left\lfloor \frac{1}{2} \left[(f_{n-1} + f_{n-2}) \right] \right\rfloor$$

$$e_n = f_n - \hat{f}_n$$

$f_1 = 21$, $f_2 = 22$, $f_3 = 27$, $f_4 = 25$, $f_5 = 22$

$f_0 = 21$

$$\hat{f}_2 = \frac{1}{2}(f_1 + f_0) = \frac{1}{2}(21+21) = 21$$

$$\hat{f}_3 = \frac{1}{2}(f_2 + f_1) = \left\lfloor \frac{1}{2}(22+21) \right\rfloor = 21$$

$$\hat{f}_4 = \frac{1}{2}(f_3 + f_2) = \left\lfloor \frac{1}{2}(\underset{13.5}{27} + \underset{11}{22}) \right\rfloor = 24$$

$$\hat{f}_5 = \frac{1}{2}(f_4 + f_3) = \left\lfloor \frac{1}{2}(25 + 27) \right\rfloor = 26$$

$$\hat{f}_6 = \frac{1}{2}(f_5 + f_4) = \left\lfloor \frac{1}{2}(\underset{11}{22} + \underset{12.5}{25}) \right\rfloor = 23$$

$e_1 = 0$
$e_2 = 1$
$e_3 = 6$
$e_4 = 1$
$e_5 = -4$

→ Difference b/w Predictive coding and DPCM is that DPCM encorporates quantizier step

→ DPCM is a ~~lossless~~ lossy conversion.

$$\hat{f}_n = \text{trunc}\left[(\tilde{f}_{n-1} + \tilde{f}_{n-2})/2 \right]$$

$$e_n = f_n - \tilde{f}_n$$

$$\tilde{e}_n = Q[e_n] = 16 * \text{trunc}\left[(255 + e_n)/16 \right] - 256 + 8$$

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

Consider $f_0 = 130$

$\tilde{e}_1 = 0$
$\tilde{f}_1 = 130$

$f_1 = 130$
$f_2 = 150$
$f_3 = 140$
$f_4 = 200$
$f_5 = 230$

130 , 150 , 140 , 200 , 230
f_1 f_2 f_3 f_4 f_5

$f_0 = 130$

$\cancel{\tilde{f}_1 = 130}$

$\tilde{e}_1 = 0$

$\tilde{f}_1 = 130$

$\cancel{\hat{f}_2 =}$

$\hat{f}_1 = 130$

$\cancel{\hat{f}_2 = \text{trun}((\tilde{f}_1 + \tilde{f}_0)/2)}$

$\hat{f}_1 = 130$

$\cancel{\tilde{f}_2 = \hat{f}_2 + \tilde{e}_2}$

$\cancel{e_n = f_n - \hat{f}_n}$

$e_1 = 130 - 130 = 0$

$\hat{f}_2 = (\tilde{f}_1 + \tilde{f}_0)/2$

$= \frac{130 + 130}{2}$

$\hat{f}_2 = 130$

$e_2 = 150 - 130 = 20$

$\tilde{e}_2 = 16 \times (255 + 20)/16$

$16 \times 17 - 256 + 8$

$= 24$

$\tilde{f}_2 = 130 + 24 = 154$

$\hat{f}_3 = \text{trunc}\ \frac{\tilde{f}_2 + \tilde{f}_1}{2}$

$\frac{\cancel{154} + 130}{2}$ 77 65

142

$\hat{f}_3 = 142$

$e_3 = 140 - 142 = -2$

$\tilde{e}_3 = 16 \times (255 - 2)/16 - 256 + 8$

$= 240 + 8 - 256$

$\tilde{e}_3 = -8$

$\tilde{f}_3 = 142 - 8$

$= 134$